Universidad Popular del Cesar

Minería de datos y Patrones

Version 2024-I

Clustering: K-means

[Capítulo 6]

Dr. José Ramón Iglesias

DSP-ASIC BUILDER GROUP

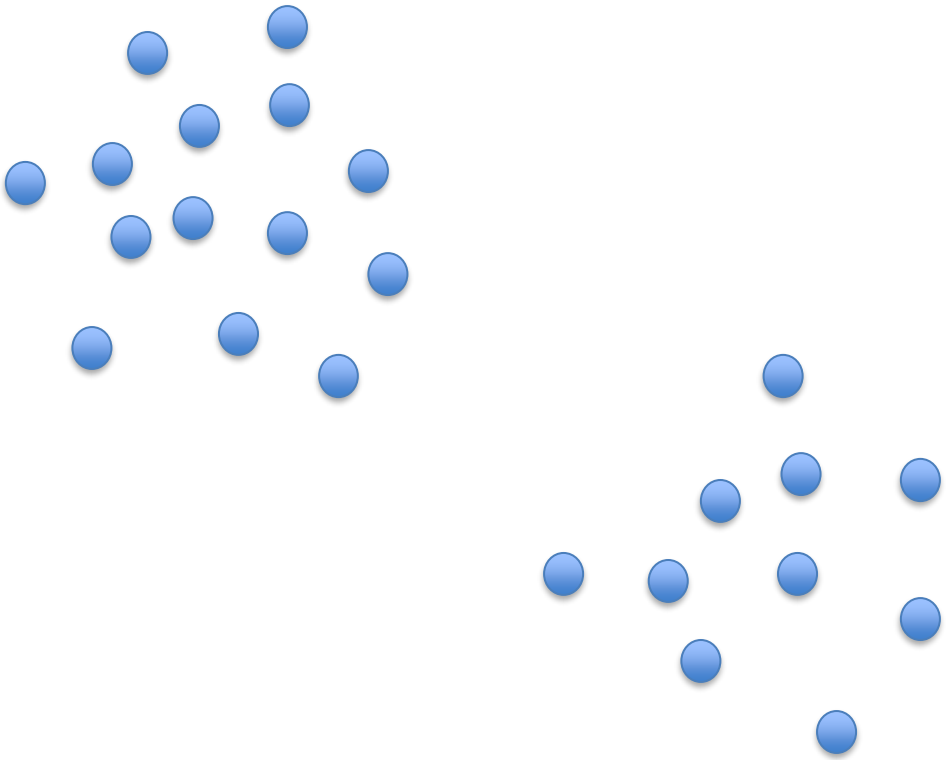
Director Semillero TRIAC

Ingenieria Electronica

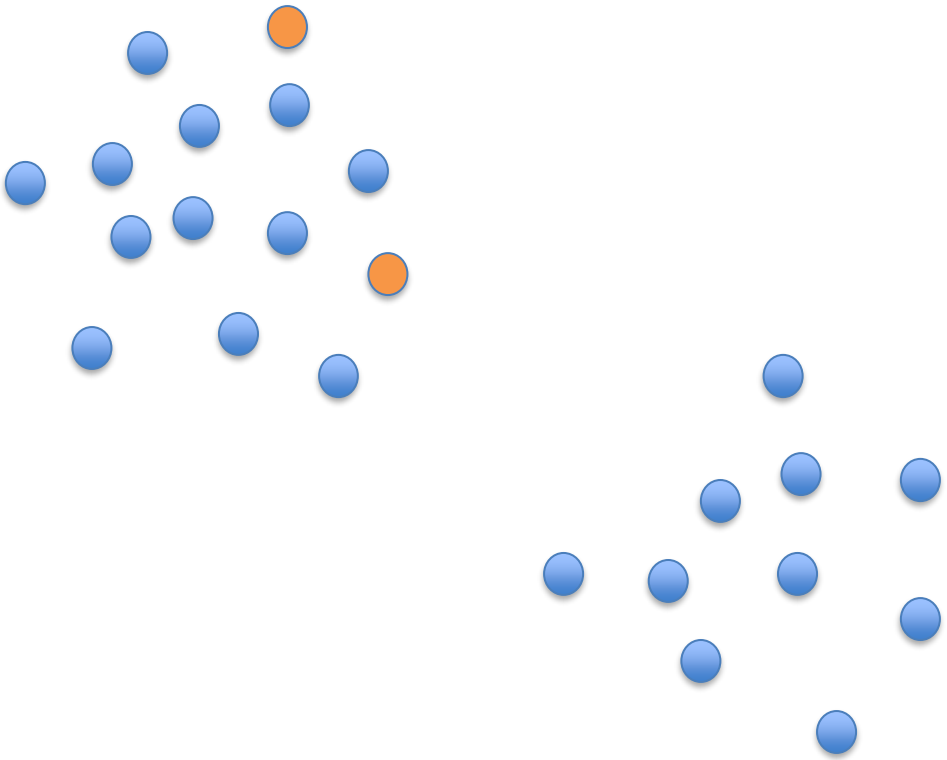
Universidad Popular del Cesar

Algorithm:

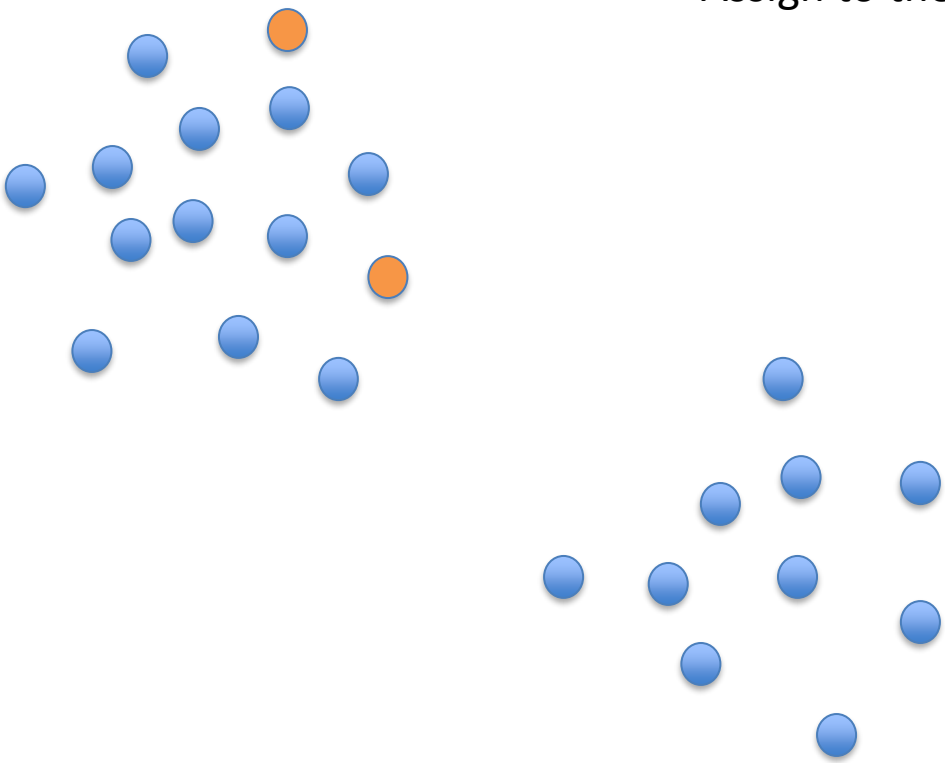
1. Input Data $X = \{x_1, x_2, \ldots x_N\}$ and number of clusters K
2. Centroids $\{c_1, c_2, \ldots c_K\}$ = random K points of X
3. For each data point x_i
4. Compute distance $d_{ij} = d(x_i, c_j)$ i=1,...,N, j=1,...K
5. Assign x_i to the nearest centroid: $y_i = \text{argmin}_j\{d_{ij}\}$
6. Compute the new centroids of each cluster
   $c^*_j = \text{mean}(x_i)$ for $y_i = j$
7. if $c^*_j \neq c_j$ then $c_j = c^*_j$ go to step 3
8. Output: $\{c^*_1, c^*_2, \ldots c^*_K\}$ and y_i for i=1,...,N

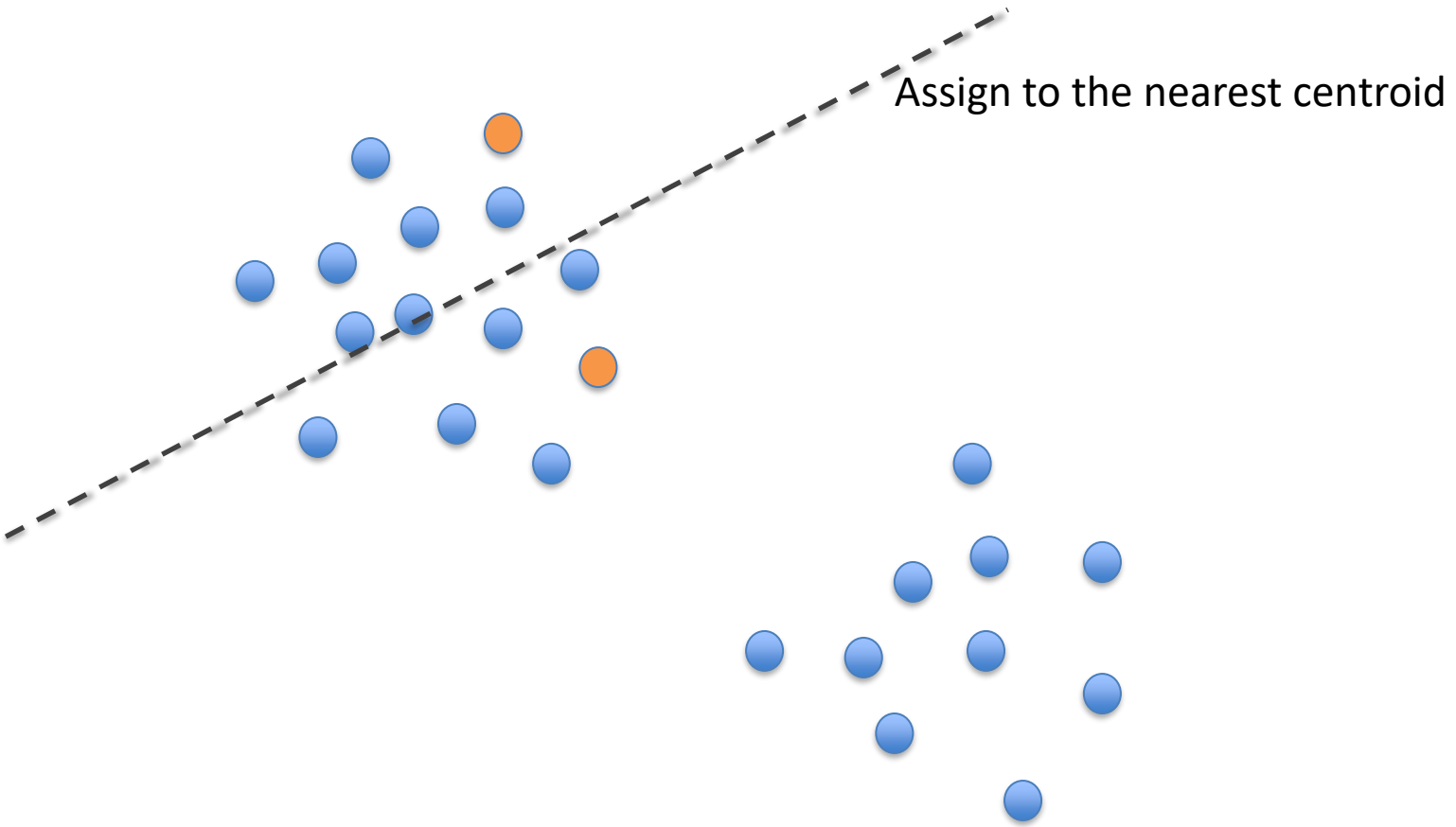


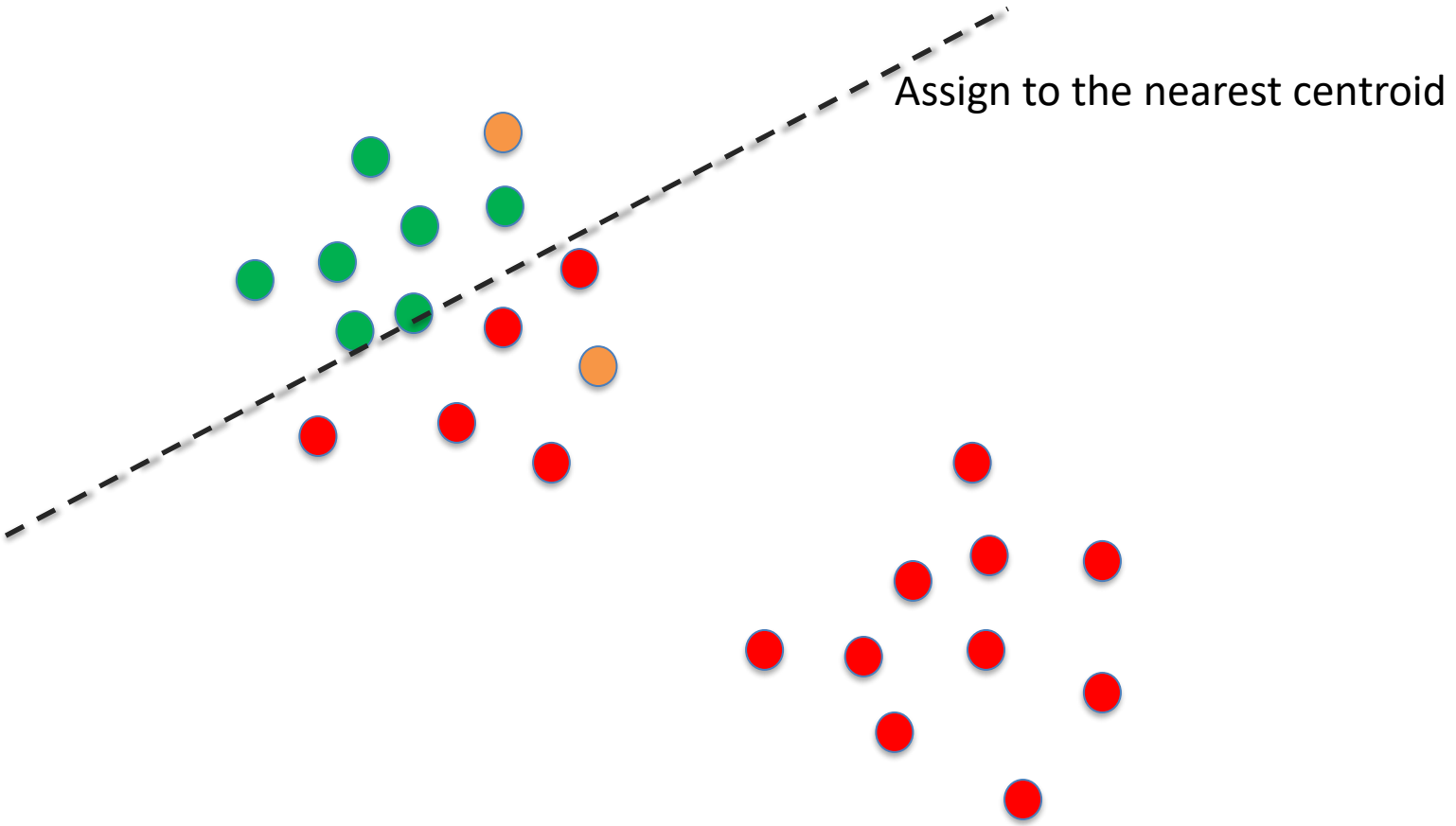
Choose random K=2 points (centroids)

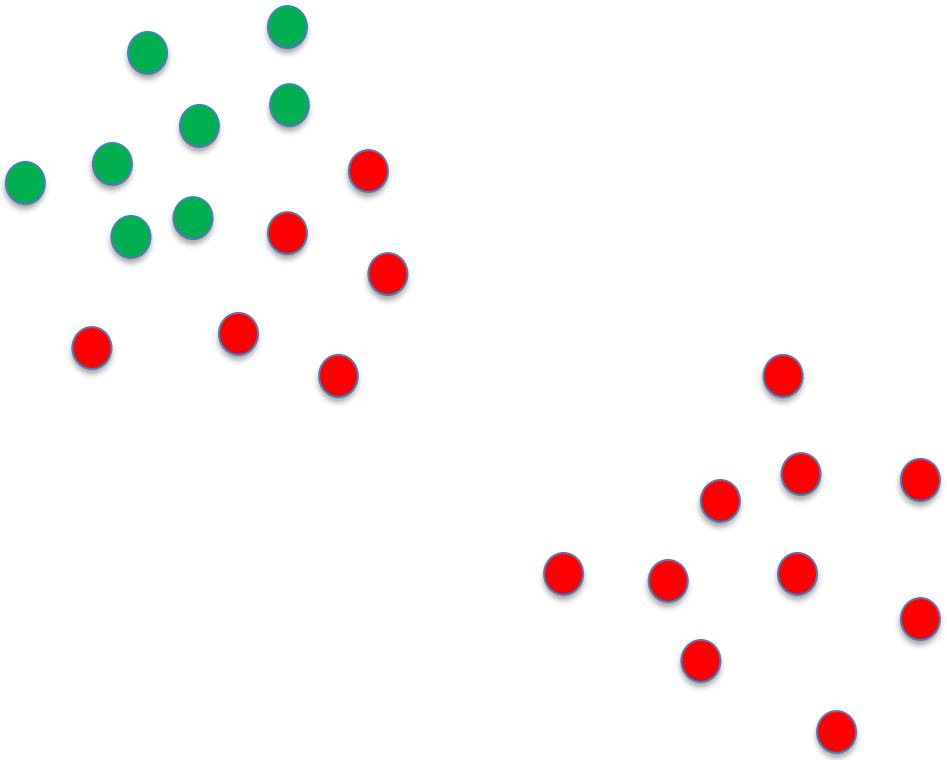


Assign to the nearest centroid

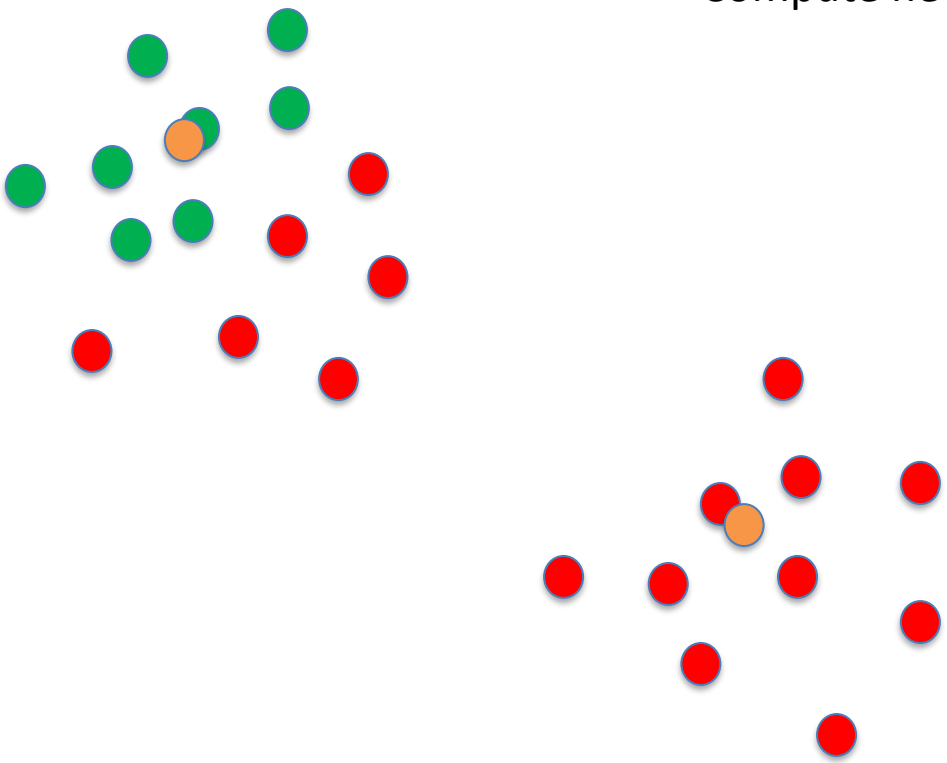


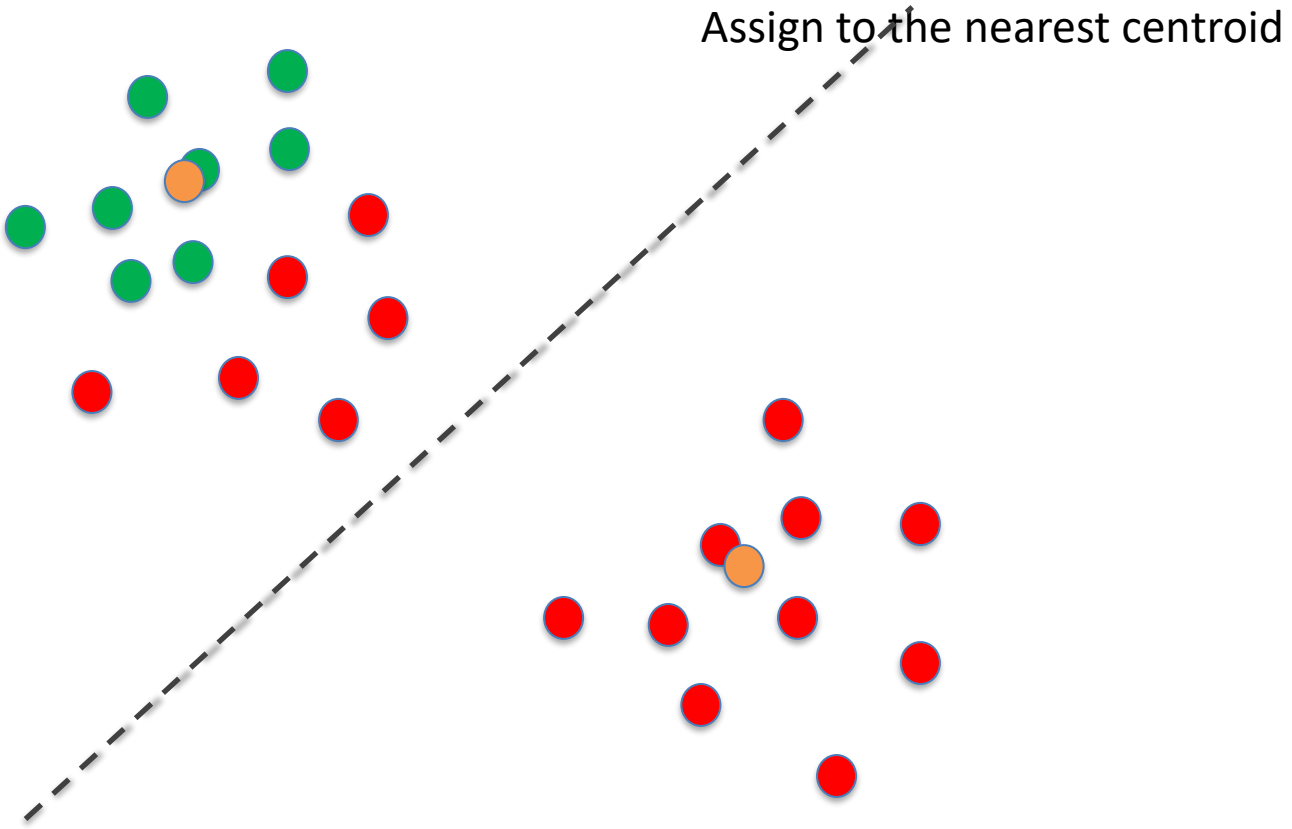

Assign to the nearest centroid

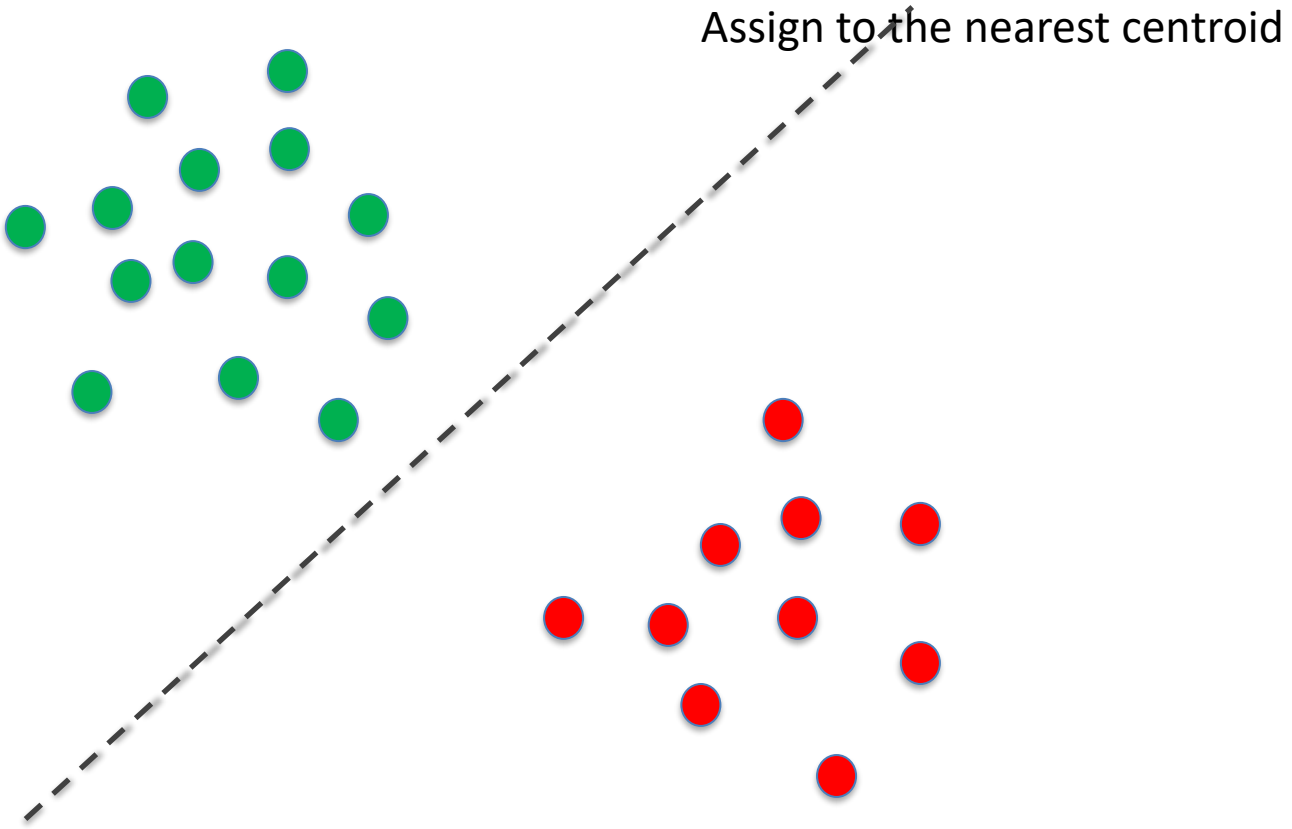

Assign to the nearest centroid



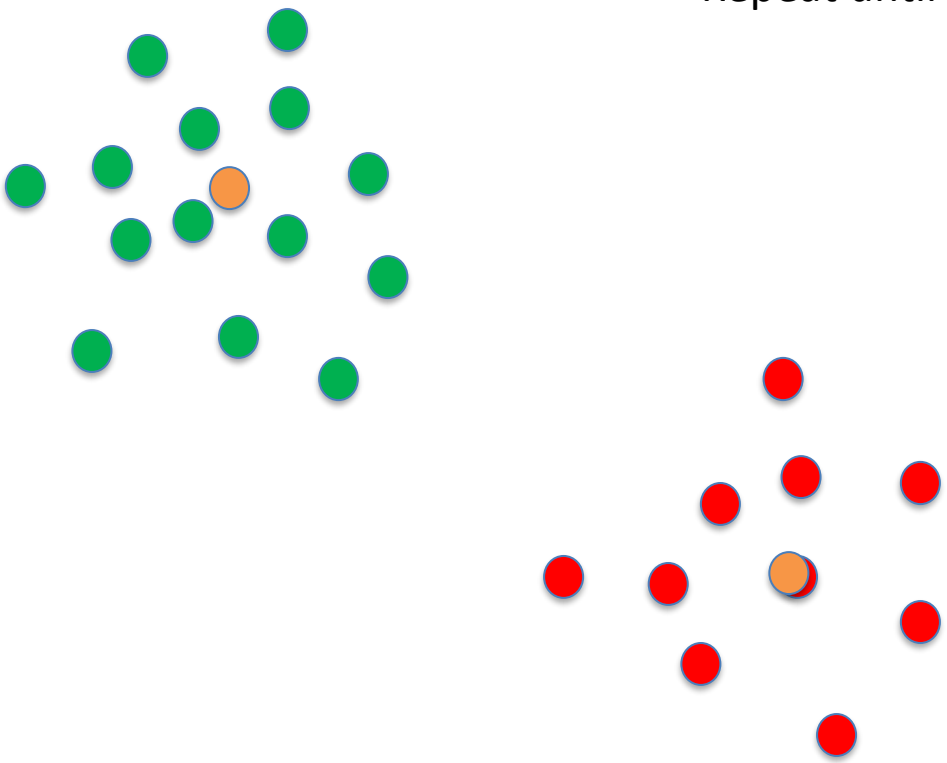
Compute new centroids




Assign to the nearest centroid


Assign to the nearest centroid

Repeat until convergence

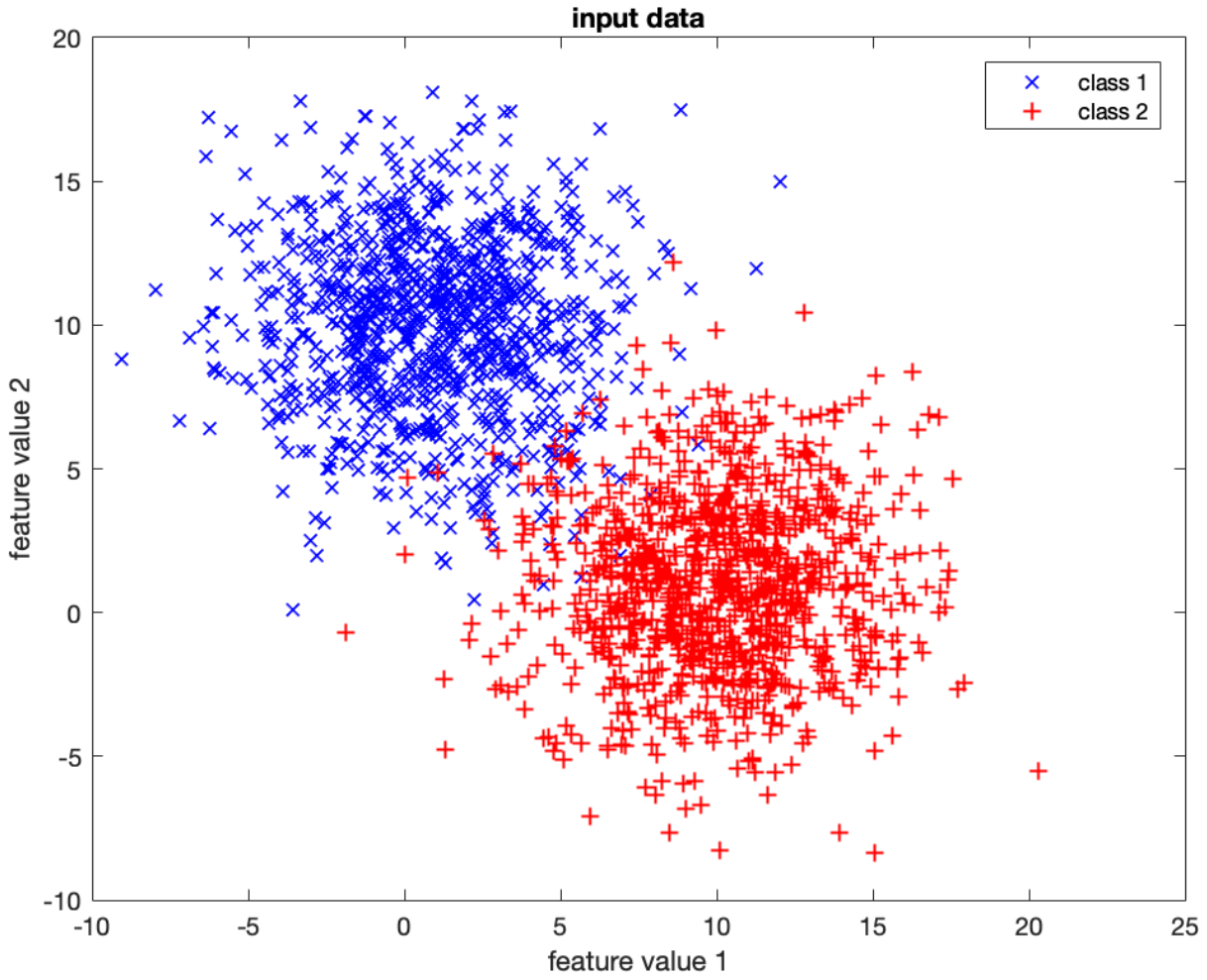


Algorithm:

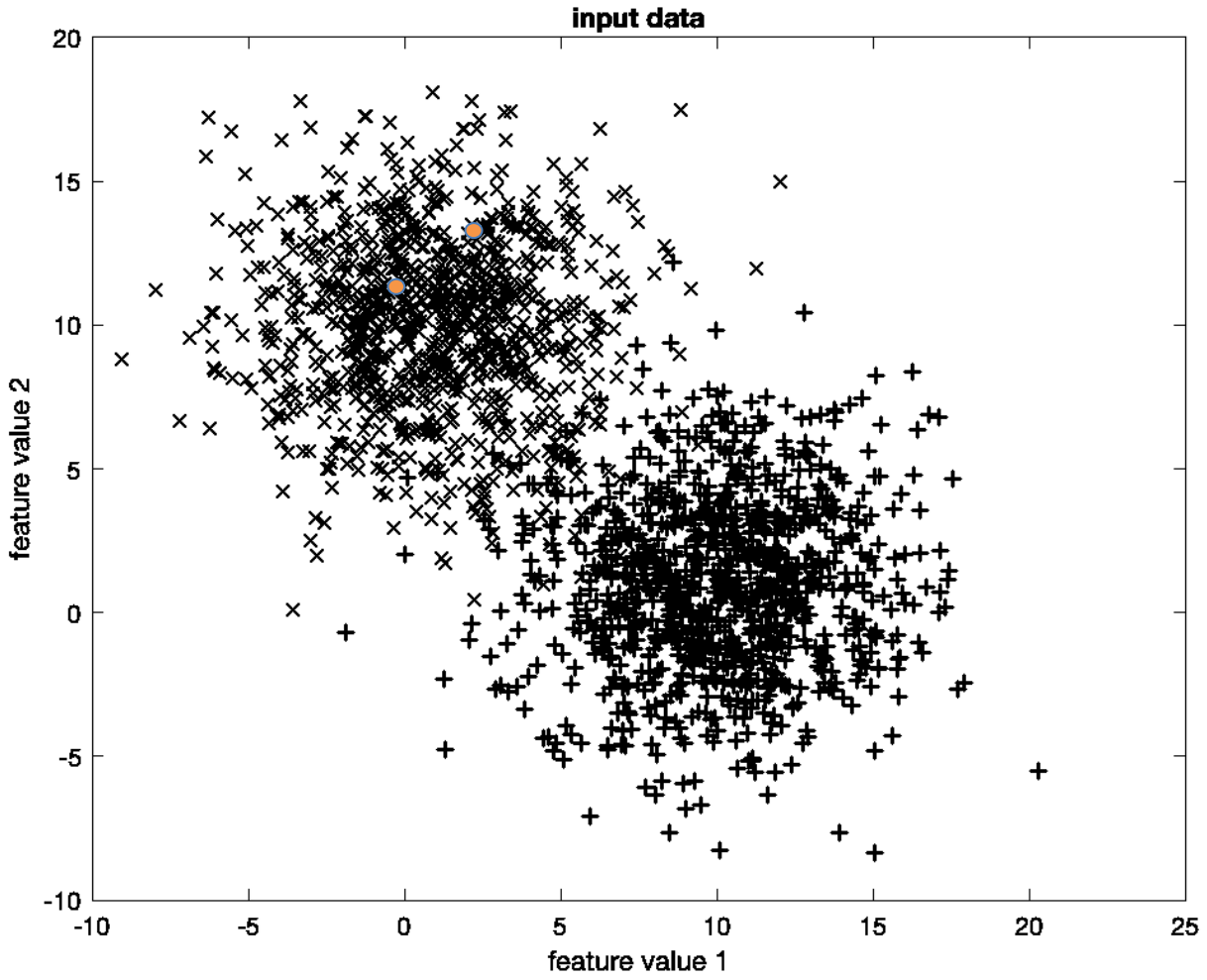
1. Input Data X = $\{x_1, x_2, \ldots x_N\}$ and number of clusters K
2. Centroids $\{c_1, c_2, \ldots c_K\}$ = random K points of X
3. For each data point x_i
4. Compute distance $d_{ij} = d(x_i, c_j)$ i=1,...,N, j=1,...K
5. Assign x_i to the nearest centroid: $y_i = \text{argmin}_j\{d_{ij}\}$
6. Compute the new centroids of each cluster
   $c^*_j = \text{mean}(x_i)$ for $y_i = j$
7. if $c^*_j \neq c_j$ then $c_j = c^*_j$ go to step 3
8. Output: $\{c^*_1, c^*_2, \ldots c^*_K\}$ and y_i for i=1,...,N

Example: Two Clouds of Points

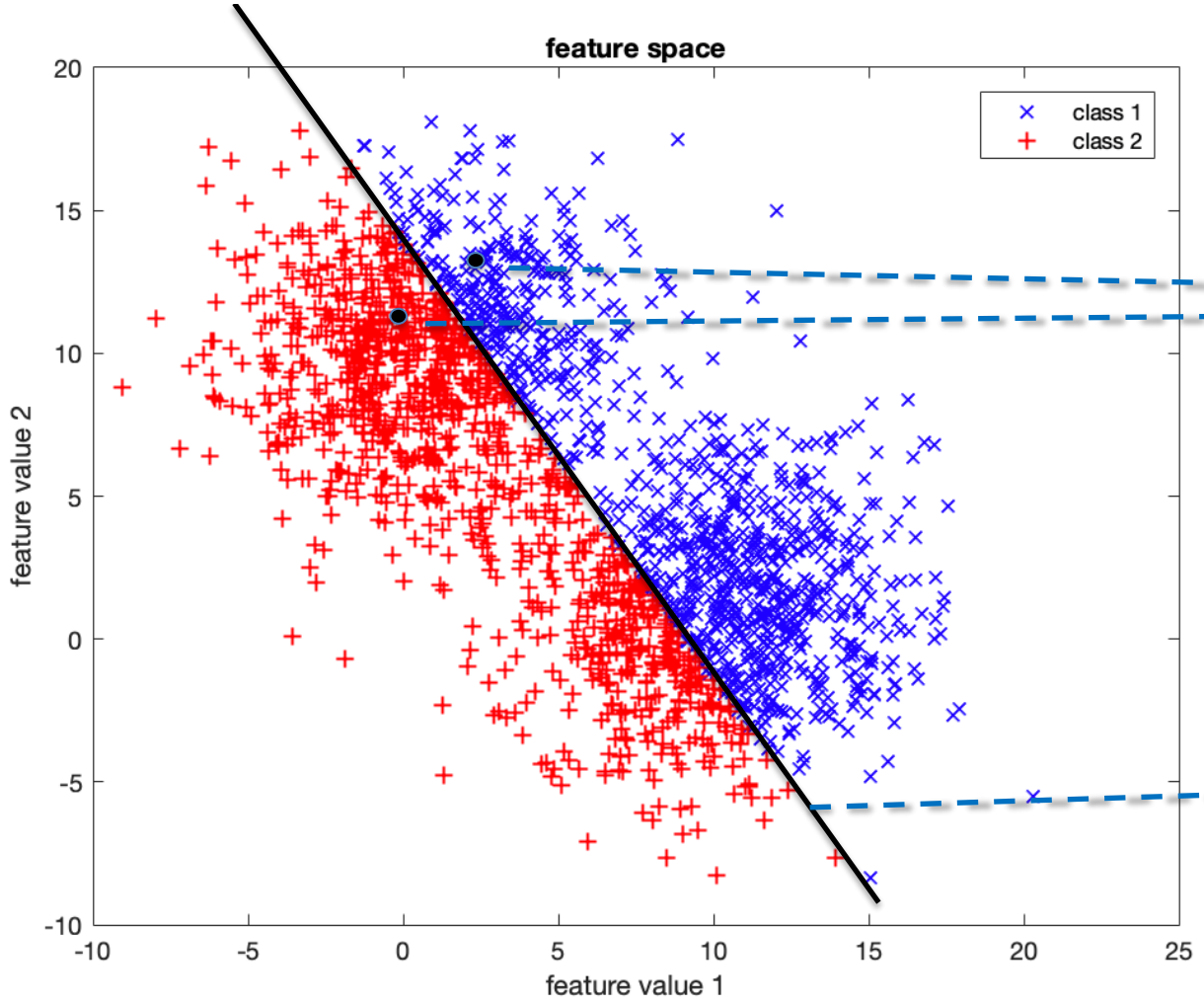
Input Data


input data
class 1
class 2
feature value 1
feature value 2
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25

Iteration 0:


input data
feature value 2
feature value 1
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25

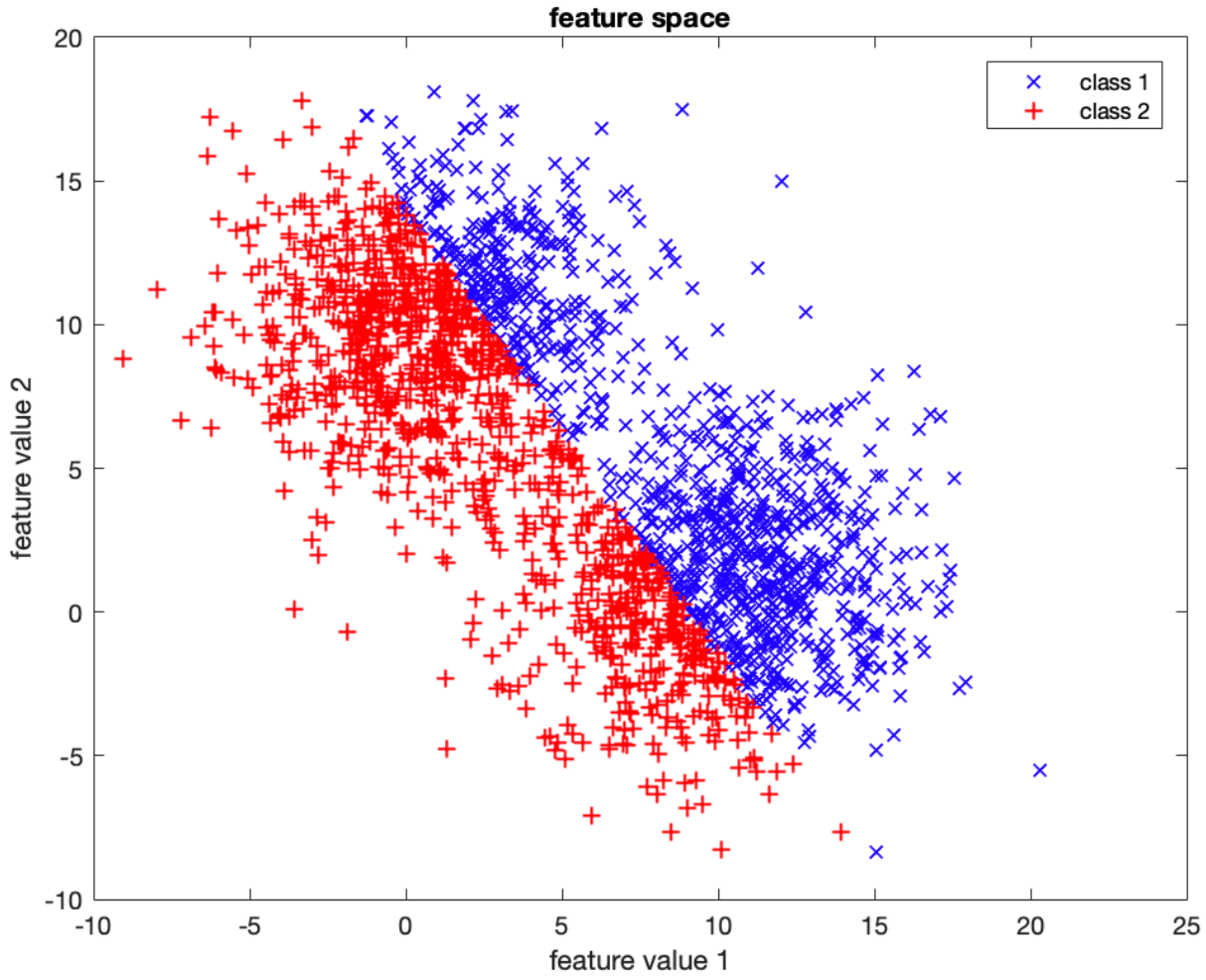
Iteration 1:



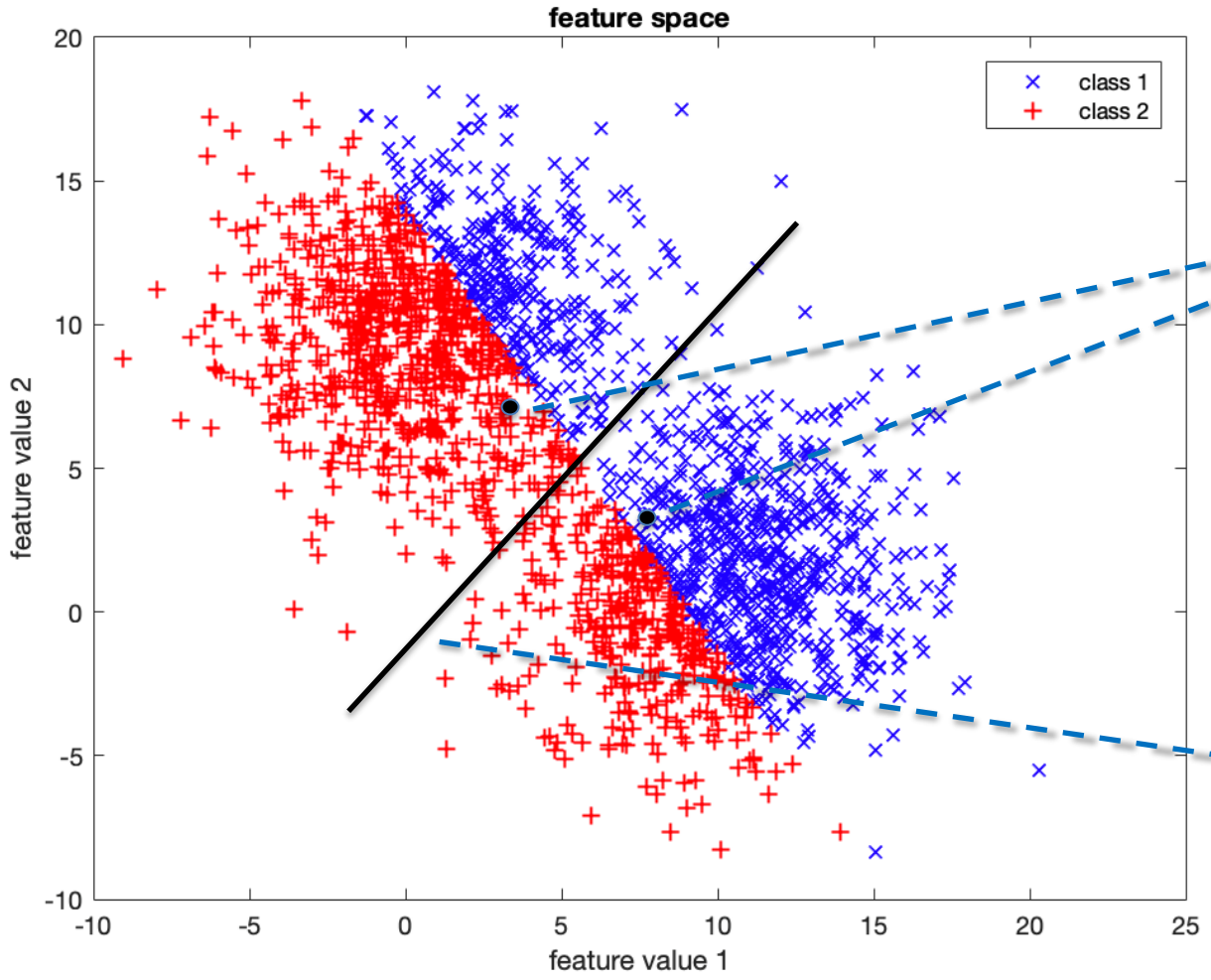
Random points

Separation line

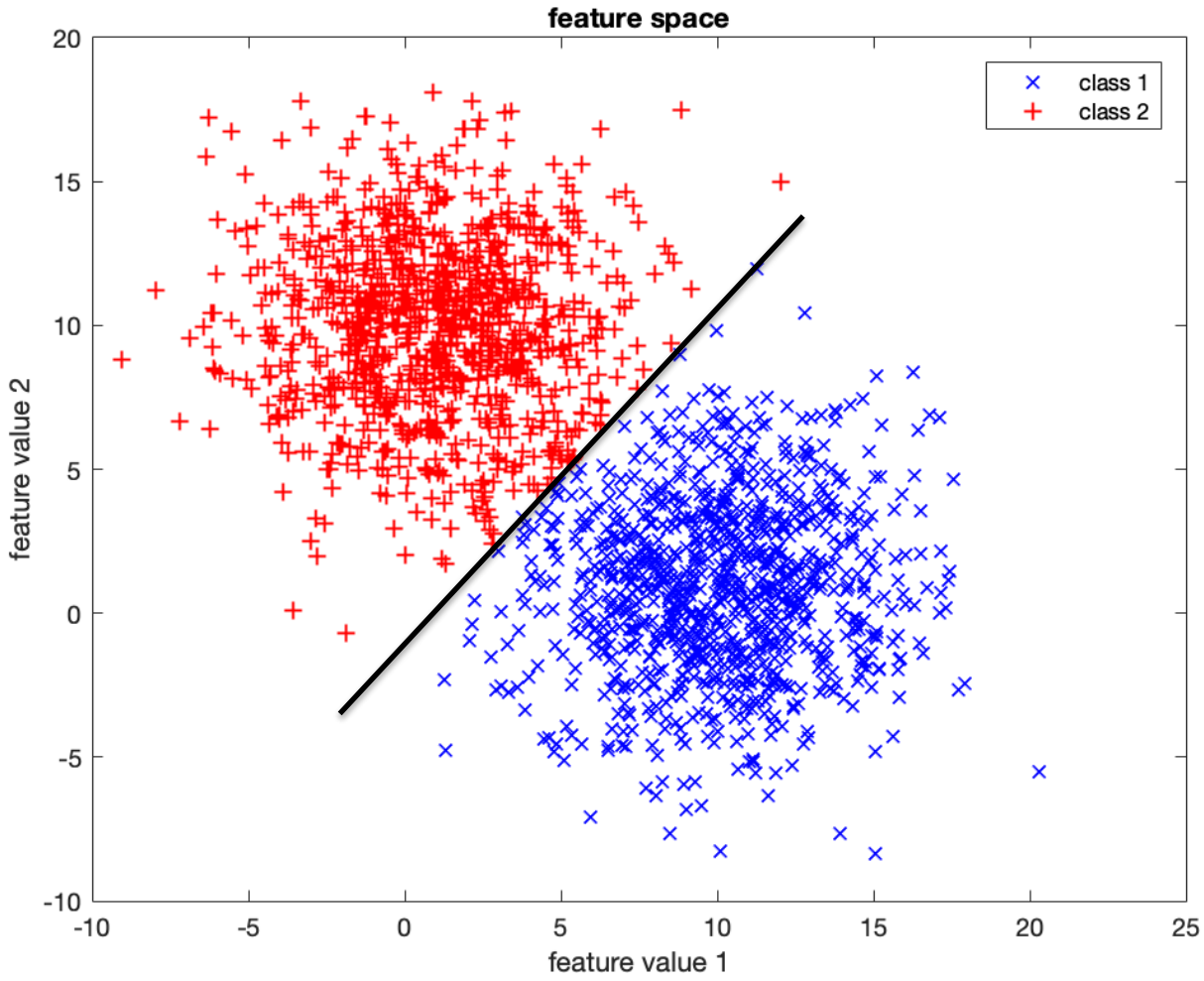
Iteration 1:


feature space
class 1
class 2
feature value 2
feature value 1
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25

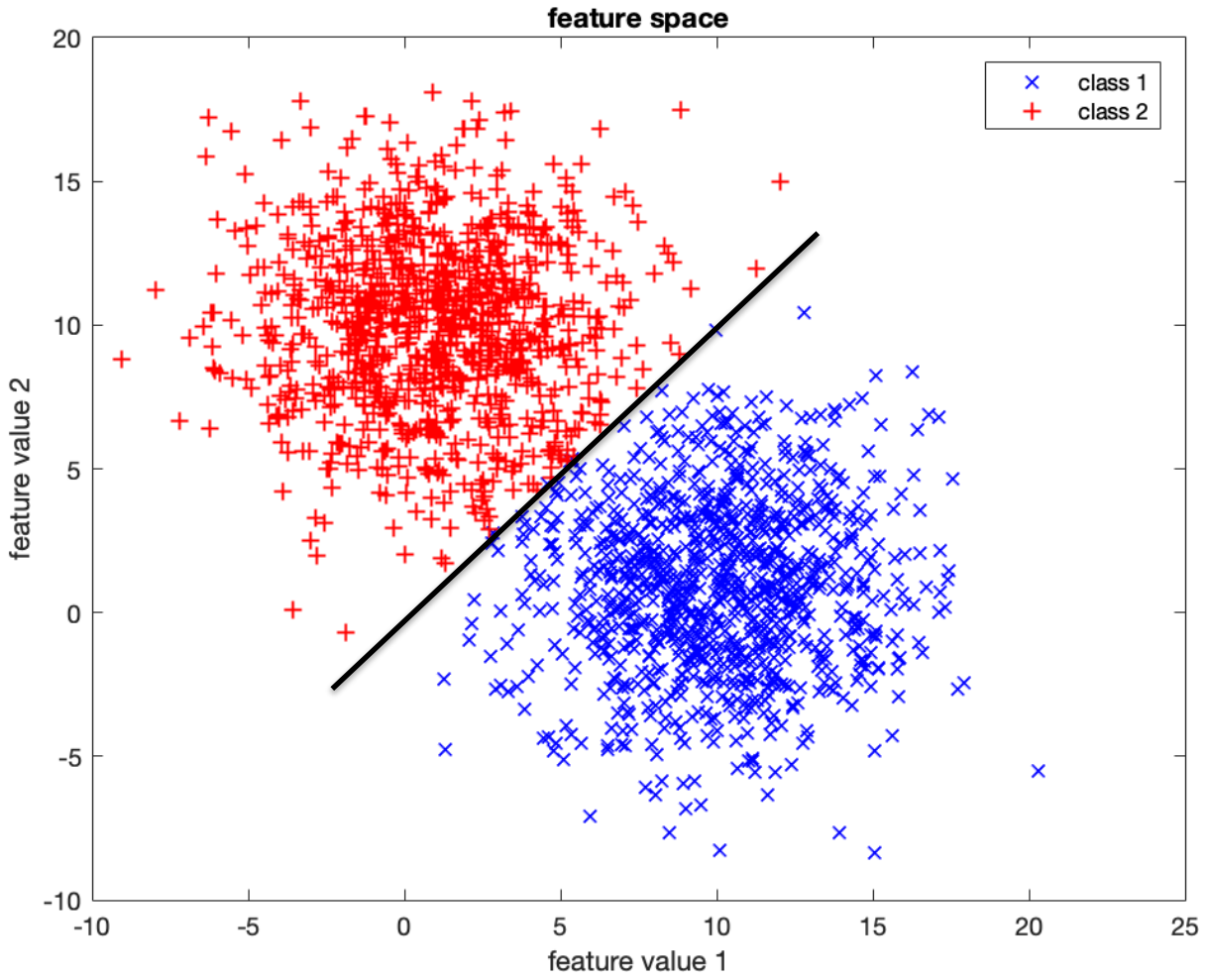
Iteration 2:


feature space
class 1
class 2
feature value 1
feature value 2
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25
Centroids
Separation line

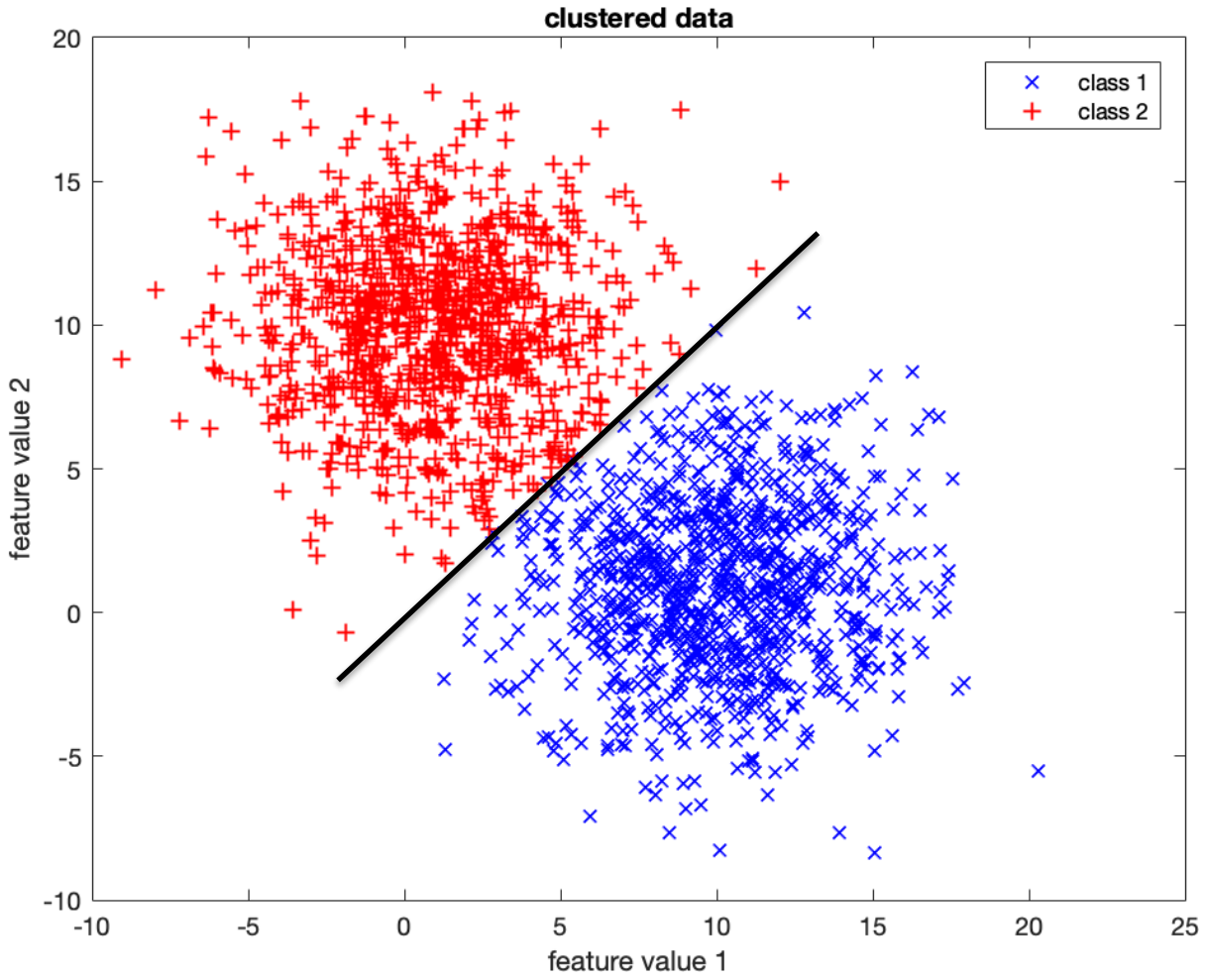
Iteration 3:


feature space
class 1
class 2
feature value 1
feature value 2
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25

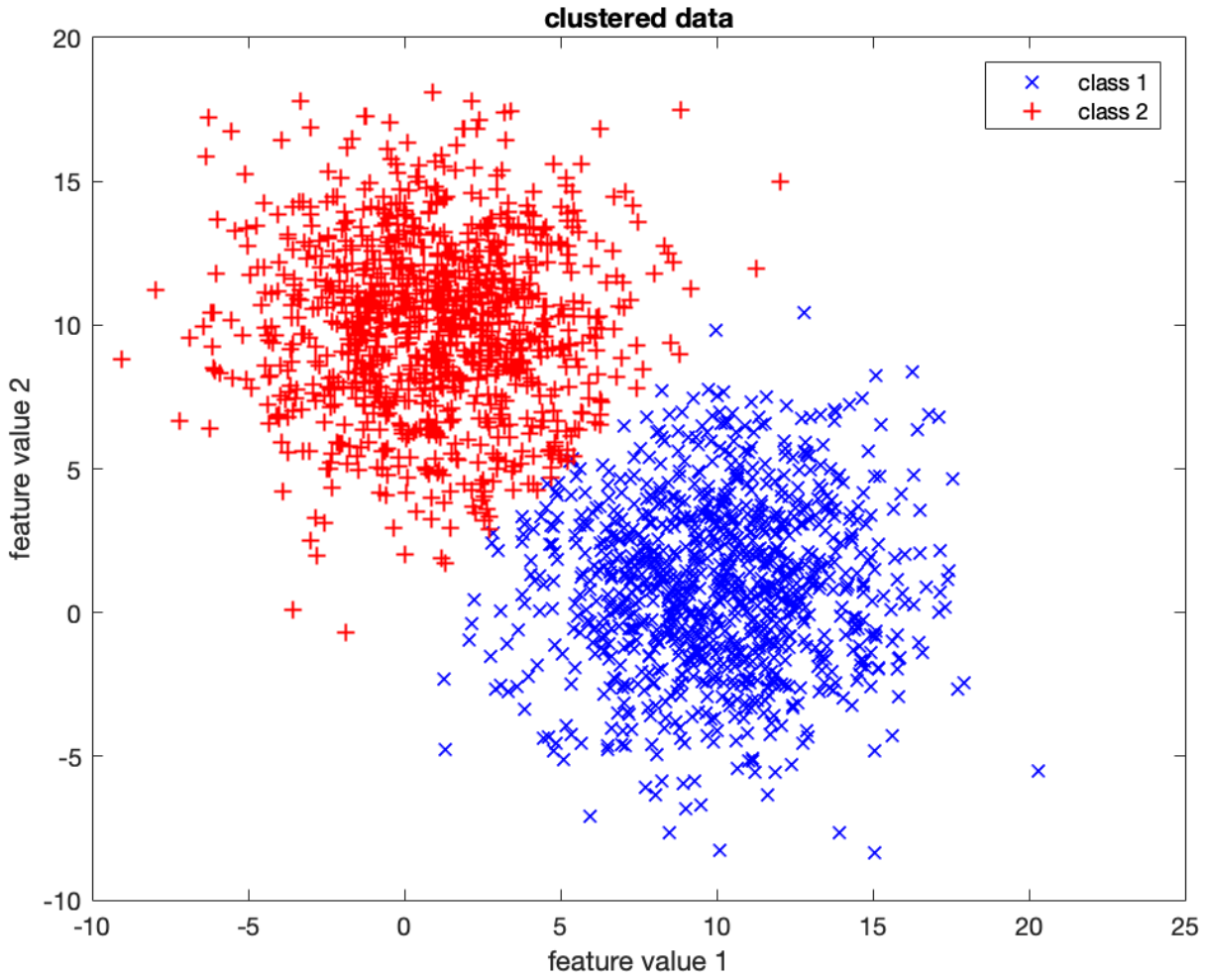
Iteration 4:


feature space
class 1
class 2
feature value 1
feature value 2
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25

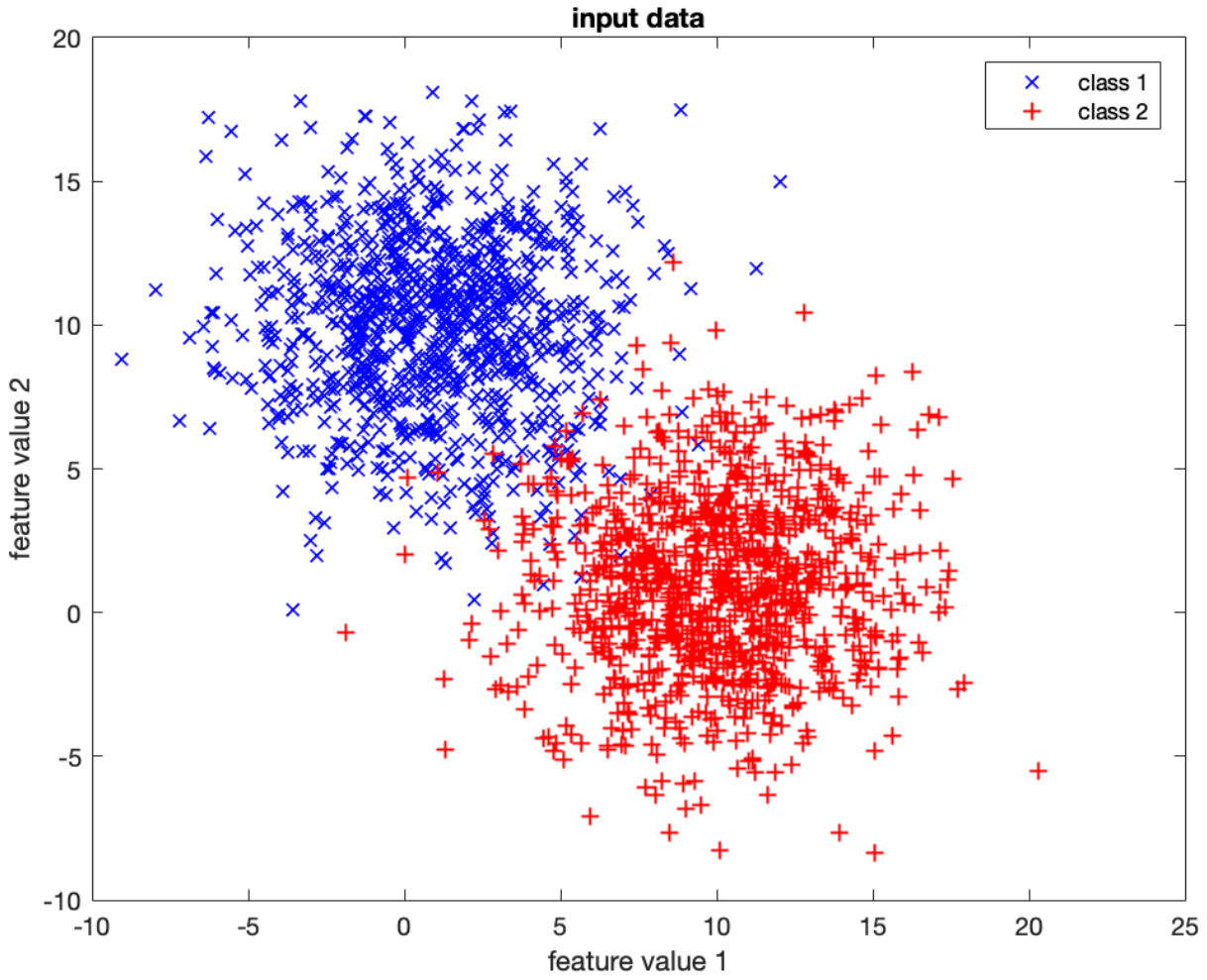
Iteration 5:


clustered data
class 1
class 2
feature value 1
feature value 2
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25

Found Clusters

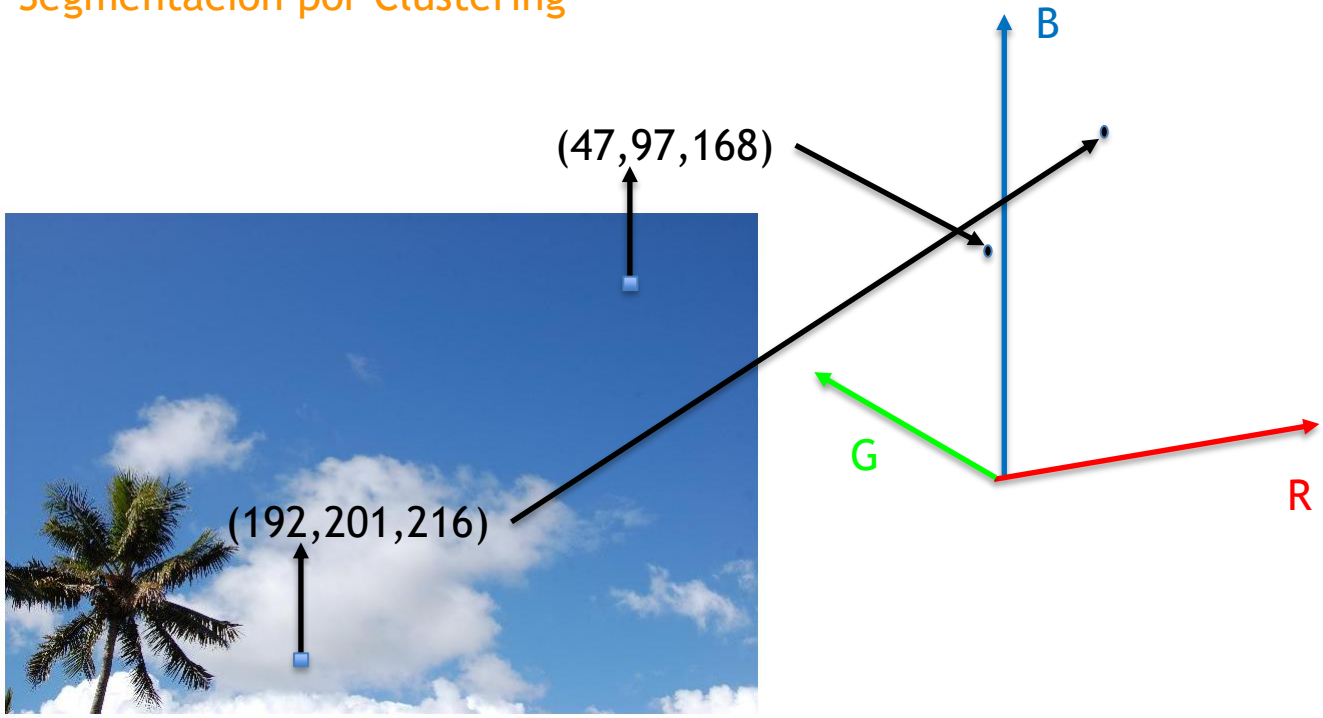


Input Data

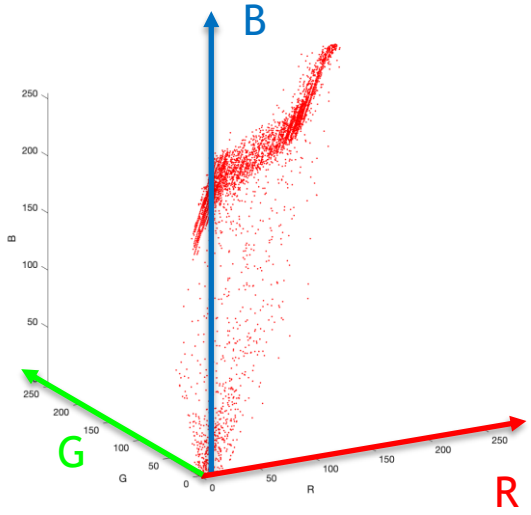

input data
class 1
class 2
feature value 1
feature value 2
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25

Example: Color Segmentation

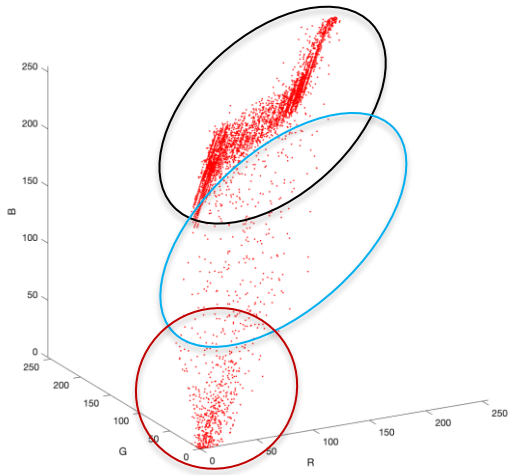
Segmentación por Clustering



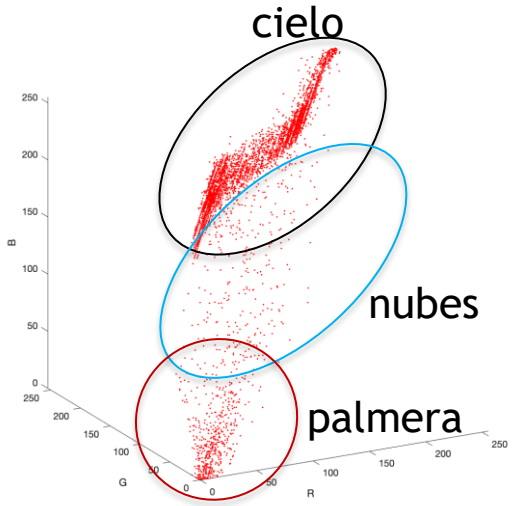
Segmentación por Clustering



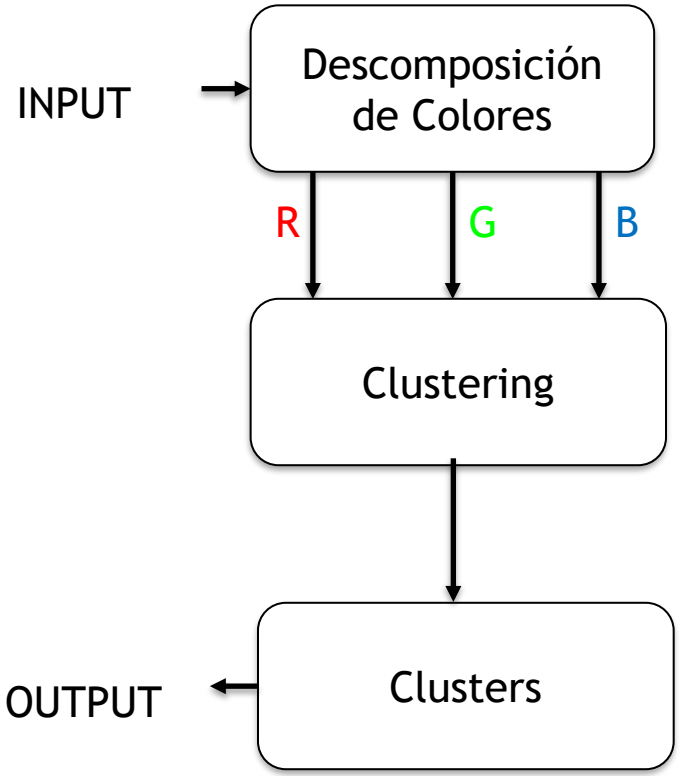
Segmentación por Clustering



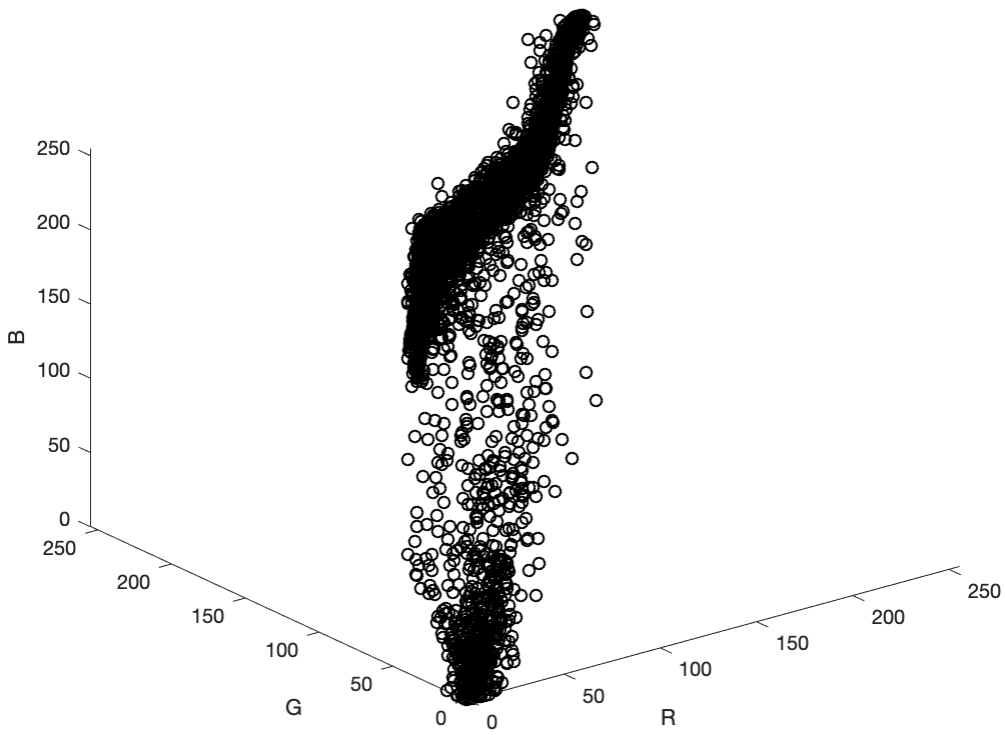
Segmentación por Clustering

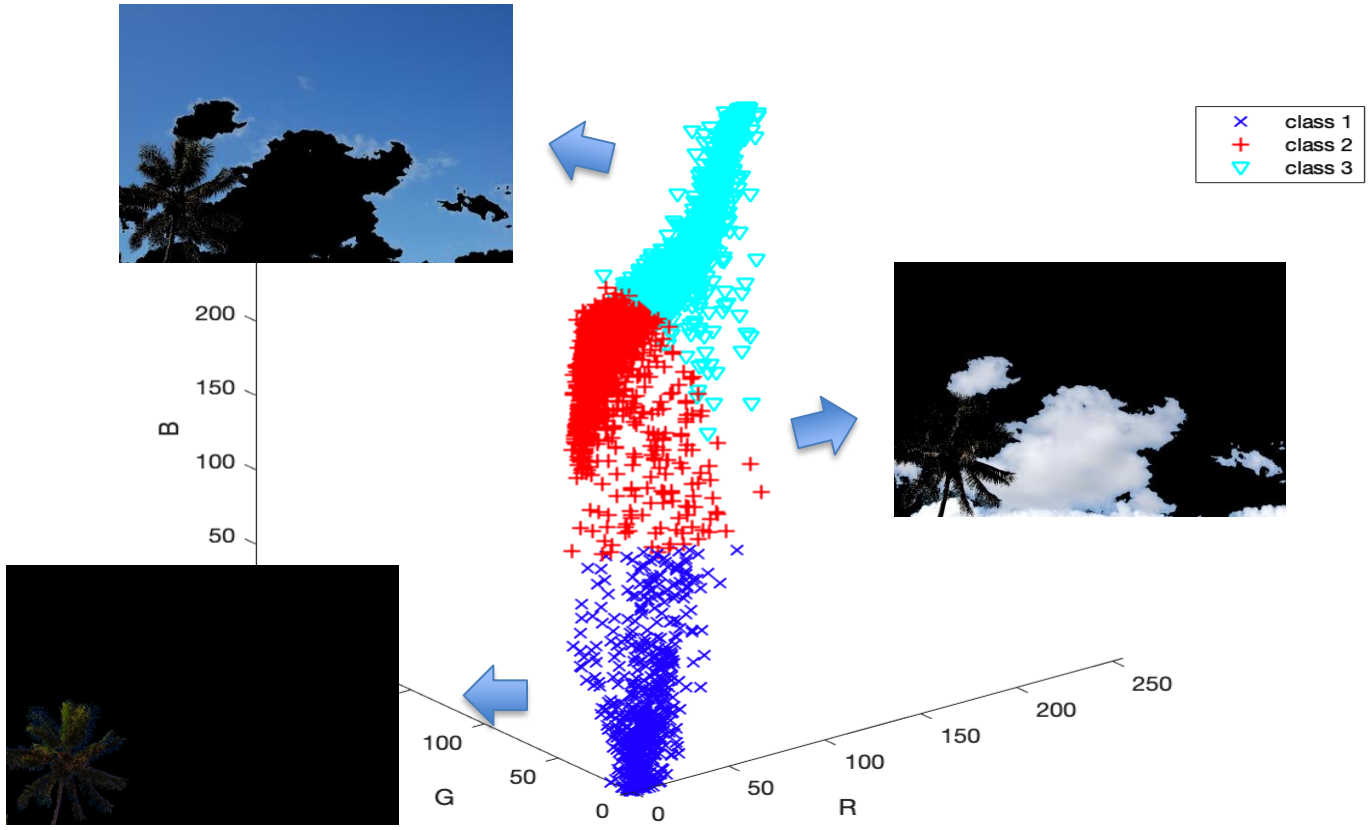


Algoritmo


INPUT
Descomposición de Colores
R
G
B
Clustering
Clusters
OUTPUT

Clustering usando k-means


B
250
200
150
100
50
0
G
250
200
150
100
50
0
R
0
50
100
150
200
250


class 1
class 2
class 3
B
200
150
100
50
G
100
50
0
0
50
100
150
200
250
R


INPUT


OUTPUT